

**A users guide for the code XPDP1, version 3.1  
and the modified version MODXPDP1. To be  
run on HP workstation 712/80 with HP-UX version  
09.03 and a digital VXT 2000<sup>+</sup> terminal.**

**by Martin Bohm**

Alfvén laboratory, Division of Plasma Physics,  
Royal Institute of Technology, Teknikringen 31-33  
S - 100 44 Stockholm , Sweden

**A users guide for the code XPDP1, version 3.1  
and the modified version MODXPDP1. To be  
run on HP workstation 712/80 with HP-UX version  
09.03 and a digital VXT 2000<sup>+</sup> terminal.**

**by Martin Bohm**

Alfvén laboratory, Division of Plasma Physics,  
Royal Institute of Technology, Teknikringen 31-33  
S - 100 44 Stockholm , Sweden

## Introduction

XPDP1 is a bounded electrostatic particle in cell code for simulating 1 dimensional Plasma Devices. The code should be run on Unix workstations with X-Windows, or PC's with X-Windows emulator. The aim of this short booklet is to make it easier to use and run the source code of XPDP1, here called pdp1.c, against a HP computer with a X-window terminal. The booklet is just a complement to the XPDP1 manuals and the user has to read these manuals parallely.

After compiling and linking the source code pdp1.c with all necessary routines one gets the executable version which is called XPDP1 since it is a X-Window version (to be run with X-Windows software). The ending c in pdp1.c means that the code is written in the programming language C. To completely understand all the statements in the code the user must be familiar with this language. However, to run the program the user doesn't have to be an expert in C. The only necessary step is to prepare the inputfile with parameters relevant for the physical problem to be simulated. The following three versions of XPDP1 exist on the HP computer with the name Triton:

1. XPDP1 version 3.0
2. XPDP1 version 3.1
3. A modified version of XPDP1 version 3.1 called MODXPDP1.

More diagnostics has been included in MODXPDP1, but the main difference between XPDP1 3.1 and MODXPDP1 is that the initial particle density distribution is optional in MODXPDP1.

Commands are written in *italics*, filenames, directorynames and keys on the keyboard are written in **bold letters**. Routines belonging to the same program are denoted by a common name written in capital letters.

**Remember** that UNIX distinguishes between capital and small letters.

## 1 A Unix session

### 1.1 How to log in

1. If the workstation is off, turn on the two electrical switches, one on the left side of the box under the screen and one on the right lower part just below the screen. Wait until a faint beep is heard from the station.
2. Press the function key **f3** and wait until a window with the head VXT V2.1 Terminal Manager pops up.
3. Move the cursor to the option Create on the upper part of the screen and press the left button on the mouse. Drag the cursor to the option xsession and chose the computer triton.plasma.kth.se and

wait.

4. Log in when the computer ask for username and password. Wait.
5. An area of different symbols will pop up on the lower part of the screen. The appearance of this 'palette' of icons will be defined by the user's login file. Click on the workstation symbol to get a window to work against. It could be convenient to have more windows, so click one more time for every more window you want. Among all the windows which will appear, the one with a highlighted frame will be active in the sense that the commands you now write will appear in this window. If you want to use another window just move the cursor to the window you want and click with the left button to make the window highlighted. Once you are inlogged you will discover that the delete key (the key on the upper right corner of the keyboard which is marked with an arrow) doesn't work in a correct way. The figure to the right of the cursor will be deleted instead of the figure to the left. The key can be made to work properly by running the program *delete*. You can copy the program from the directory `/net/triton_disk2/user/bohm` if you don't have the program your directory. The program is ran by giving the command *delete*. More about the program *delete* in chapter 1.7.

## 1.2 How to log out

1. Click on the exit symbol on the lower right of the screen.

## 1.3 Hints

An exhaustive manual for the HP computer is available on the computer named Luna. Log in to this computer by the command *rlogin luna* (remote login). The command *lrom* will then make the manual available. The manual is self instructive and gives instructions how to continue but remember to look for the HP 700 serie because the manual covers also several other computers. If you want you can put the command *lrom* in background by adding the ending *&* (*lrom&*). A job in background doesn't need any window.

Very useful is also the selfinstructive window denoted Helpview Help which is shown on the screen when you log in. The way the terminal session is started up when you have logged in can be set by clicking on the icon with the following symbols on it: a palette, three letters of T, a small rectangel and a corner. After clicking, a window with symbols will pop up. Click on the key symbol and a list of options will appear. If the option denoted: Resume current session is marked, all the settings and all the windows which was open when you logged out will reappear when you log in again. This can cause problem when you log in if too many processes and too many windows were opened at the earlier session. This problem may not appear if the option: Return to Home session, is chosen.



## 1.4 The directory tree

The files in Unix are sorted under a directory tree, fig. 1. After you have logged in you will be in your home directory. If you want to see in what directory you are give the command *pwd*. To see the contents in the directory you are in, give the command *ls*. This command can be given with different flags: *ls -l* gives more details about the files e.g. dates when they are written, *ls -a* lists all files even filenames beginning with '.' and *ls -t* lists the files in order of creation. You can move in the directory tree by the command *cd* with an appropriate flag or by just giving the name of the directory you want to go to:

<i>cd</i>	moves the user to his homedirectory
<i>cd ../.</i>	moves the user two directories up
<i>cd ..</i>	moves the user one directory up
<i>../directoryname</i>	moves the user one directory level up then down to the directory named <i>directoryname</i> .
<i>cd directoryname</i>	moves the user to the directory <b>directoryname</b> . As an example, <i>cd /net/triton_disk2/user/bohm</i> moves the user to the directory <b>bohm</b> .

The following files, directories and subdirectories exist under the directory **/net/triton\_disk2/user/bohm**:

**delete** Running this file makes the delete key to work properly

**Mail** A directory containing all the mail.

**pdp1source** Contains the sourcefiles of XPDP1 version 3.0.

**xpdp2** Contains three subdirectories: **modxpdp1** which contains the modified version MODXPDP1, **xpdp1** which contains XPDP1 version 3.1 and **xgrafx** which contains the graphics belonging to XPDP1. The directory also contains the tarfile **xpdp1.tar** (see below) and the textfile **README**.

**xpdp1exe** which contains copies of the executable xpdp1 version 3.0 with inputfiles (copied from the directory under the username torven in **/net/triton\_disk2/user/torven**

**xpdp1backup** contains a backup of XPDP1 version 3.1.

## 1.5 How to make a sourceprogram executable in Unix

A program has to be linked and compiled before it can be executed. The executable program is runned by writing its name. The compiling process which translate the sourcefile into an objectfile is made with the command `cc` with appropriate flags (give the command `man cc` for more information). The objectfile has to be linked with appropriate libraries before an executable is created. Suppose you have written a program called **sourceprogram.c**.

The command

```
cc -o a.out -lX11 -L/lib -lm -L/usr/lib sourceprogram
```

will then both compile and link the sourceprogram with the resulting executable file called **a.out**.

The files which have to be linked with the objectfiles in this case are the files in the directory `X11`.

The letter *L* in *L/usr/lib* means that the `X11` files is not located in the default library

`/usr/include/X11` but are in this case situated in the library `/usr/lib/X11`.

The use of a makefile simplify the compiling and linking process considerable. A makefile is a file containing the necessary commands for the compiling and linking processes. By giving the command *makefile* the commands in the file are executed and the compiling and linking processes is made for all files which have be changed since the last use of makefile.

## 1.6 Some Unix commands

If you want to know more about a command, write *man* or *man -k*. followed by the command you want to know more about.

command

<code>cp filename newfilename</code>	Copies a file named filename to newfilename
<code>cp filename1 filename2.....directoryname</code>	Copies the files filename1, filename2 etc to the directory directoryname
<code>cp -r directory newdirectory</code>	Copies the everything under the directory to newdirectory
<code>mv name newname</code>	(move) Changes the file/directoryname name to newname
<code>mv filename1 filename2.... directoryname</code>	Moves filename1 filename2 etc to the directory directoryname
<code>rm filename</code>	(remove) Deletes files
<code>ls file/directory name</code>	(list) Lists the file filename or the contents of the directory directoryname

<i>ln -s directory/filename linkname</i>	Creates the symbolic link linkname to the directory or file directory/filename. Giving the command linkname will then move you to the directory or file directory/filename.
<i>mkdir directoryname</i>	(make directory) Creates the directory directoryname
<i>rmdir directoryname</i>	(remove directory) Deletes the directory directoryname which must be empty
<i>rm -r directoryname</i>	(recursive) Deletes the directory directoryname
<i>rm -i</i>	Deletes files interactively
<i>cat filename</i>	Lists the contents of filename on the screen
<i>more filename</i>	Lists the contents of filename on the screen, one screenpage each time you press the space bar and one line each time you press the return button. With more you can also search in the file
<i>cd directory</i> (cd with different prefixes as .. and ../.. etc gives the path to the new directory)	(changes directory)
<i>who</i>	Lists all the inlogged users/processes together with the name of the terminals.
<i>ps</i>	Lists started processes
<i>ps -t termname</i>	Gives the identification number (pid) of the terminal with the name termname
<i>ps -ef</i>	Gives all the processes started by the user
<i>kill -9 pid</i>	Kills the process with the identification number pid
<i>bdf</i>	Gives the total disk memory used. Up to 75% of the total space can be used by a user (950209) without causing to much problems.

*du*

6

Gives the user's disk memory used in each directory in blocks (1 block is 512 bytes). Can be used with different flags, give the command *man du* for more information.

*du -s \$home*

*\$home* is a shell variable. A shell variable is a variable which is used by the computer to search for the file corresponding to the command given by the user.

*echo \$home*

Gives the value of the variable *\$home*.

*env*

Shows all the variables set in the shell variable *PATH*.

*renice -n 10 processnumber*

Increasing the priority number of a process by 10 means that the priority will be decreased by 10. Replace the word *processnumber* by the number of the process. Set a lower priority to a timeconsuming process. Decreasing the priority by 10 can be a good start. Observe that a decrease in the priority (increase in the priority number) for a timeconsuming job will not make life significantly better for other jobs.

*ps -fel*

Gives the priorities of the different processes. The normal priority number is 20 which is interpreted as 0 by the computer. If you lower your priority by 10 by increasing the priority number from 20 to 30, the computer will interpret the new number as 10. The lowest priority number is 39 (interpreted by the computer as 19).

## 1.7 More useful commands

If the commands in a file are executable, they are executed when the name of the file is given as a command. The file must be made executable before this. A file with the name *filename* is made executable by the command *chmod x filename* (the command *man chmod* and reference 2 give more information about the command *chmod* used with different flags).

As an example the file `delete` in `/net/triton_disk2/user/bohm` contains the lines:

```
#!/bin/ksh (which means that the file is executed in a program called
            kornshell)
stty intr ^Y (sets the terminalemulators interrupt sign to ctrl-Y)
stty erase ^? (sets the terminalemulators erase sign to ctrl-?)
```

Ctrl-Y means the sign created when both the ctrl key and the letter y key are pressed at the same time.

When `delete` is given as a command these lines are executed with the result that the delete key can be used in the rest of the terminalsession.

The variable `PATH` contains information where the computer should search when one gives a command (see ref. 2). More information about shell programmes is given in ref. 2.

A shell variable is set by the command `variablename=value` followed by the command `export variablename` where variable is the name of the variable and value its new value.

The command `env` shows all the variables set in `PATH`

A useful command is `grep` which search for a string in a directory, e.g. `grep string filename` search for the string `string` in the file `filename`, `grep string *`, search for the string `string` in all the files in the directory.

## 2 The code XPDP1

### 2.1 Introduction

XPDP1 is a bounded electrostatic code for simulating 1 dimensional Plasma Devices. It is written in C for Unix workstations with X-Windows. Three versions of XPDP1 exist on the HP computer:

1. The original version with the sourcefiles saved in the directory `/net/triton_disk2/user/bohm/pdp1source` and the original executable file with the inputfiles saved in `/net/triton_disk2/user/bohm/xpdp1exe` (and in `/net/triton_disk2/user/torven/xpdp1`).
2. The latest version of XPDP1 saved in the directory `/net/triton_disk2/user/bohm/xpdp2/xpdp1`.
3. The modified version of the latest XPDP1 version saved in the directory `/net/triton_disk2/user/bohm/xpdp2/modxpdp1`.

A list of all the routines which should be compiled and linked together to make XPDP1 3.1 executable is given in appendix 7.



## 2.2 The creation of MODXPDP1 from XPDP1 3.1

The source code `pdp1.c` and some of the routines of XPDP1 version 3.1 has been modified to enable a nonuniform initial spatial particle distribution to be simulated. The modified version of XPDP1 3.1 is called MODXPDP1. In running MODXPDP1 the user has to decide if the initial particle density distribution at  $timestep = 0$  should be uniform or nonuniform. The decision is made by setting the flag `densprofile` in the input file to 0 for a uniform or to 1 for a nonuniform initial density distribution. If one decide to have a nonuniform density distribution, a function describing the initial density distribution in space has to be given in a new subroutine. The routine is written only for symmetric maxwellian distribution but can easily be modified to include other distributions. The routine has to be added to the other routines in the compiling and linking process. Routines with different initial distribution has been given different names, but all have names beginning with `den` and are saved in the directory

`/net/triton_disk2/user/bohm/xpdp2/xpdp1/newfiles`. In the modified code it is also possible to simulate particles in the simulation region to be lost with a prescribed probability. The parameter `lossprob` (also read from the input file) is used in the subroutine `modpadjus.c`. This parameter determines the probability for a particle to disappear from the simulation region. If `lossprob` is set to 0 no ionlosses will be introduced.

A list of the files in this directory is given in appendix.

Some minor changes from the original code `pdp1.c` has also been made with the diagnostic tools. Three probes measuring the time variation of the electric field at given space coordinates and a probe measuring the timevariation of the electric potential has also been incorporated in the program as well as plots of the velocity distribution function at the space coordinate given in the inputfile.

## 2.3 How to handle `xpdp1.c`

The source code `pdp1.c` is most easily compiled and linked by the makefile which was distributed with the `pdp1` programpackage. Writing the command `makefile` (if you have the makefile in your directory) compiles and links `pdp1.c` and the accompanying routines (all saved in the directory `/net/triton_disk2/user/bohm/xpdp2/xpdp1`) The makefile does the linking process (or compiling process) only if any of the source files (or objectfiles) has been changed since the last use of the makefile.

Let us suppose that the name of the executable program of the source program `pdp1.c` is `xpdp1` (The name is given in the makefile). Observe that an executable program called `xpdp1` (version 3.0 of XPDP1), exists somewhere in the directory tree and the command `xpdp1` would hence run this version.

Once you are in the directory with the executable program `xpdp1`, you can run the program by writing its name followed by the input file: `xpdp1 inputfile` where *inputfile* is the file (usually with the extension `.inp`) from which the main program read all the inputparameters. This file doesn't have to be compiled but one can just change the parameters in the file using an editor and the

<i>du</i>	Gives the user's disk memory used in each directory in blocks (1 block is 512 bytes). Can be used with different flags, give the command <i>man du</i> for more information.
<i>du -s \$home</i>	<i>\$home</i> is a shell variable. A shell variable is a variable which is used by the computer to search for the file corresponding to the command given by the user.
<i>echo \$home</i>	Gives the value of the variable <i>\$home</i> .
<i>env</i>	Shows all the variables set in the shell variable <i>PATH</i> .
<i>renice -n 10 processnumber</i>	Increasing the priority number of a process by 10 means that the priority will be decreased by 10. Replace the word <i>processnumber</i> by the number of the process. Set a lower priority to a timeconsuming process. Decreasing the priority by 10 can be a good start. Observe that a decrease in the priority (increase in the priority number) for a timeconsuming job will not make life significantly better for other jobs.
<i>ps -fel</i>	Gives the priorities of the different processes. The normal prioritynumber is 20 which is interpreted as 0 by the computer. If you lower your priority by 10 by increasing the priority number from 20 to 30, the computer will interpret the new number as 10. The lowest priority number is 39 (interpreted by the computer as 19).

## 1.7 More useful commands

If the commands in a file are executable, they are executed when the name of the file is given as a command. The file must be made executable before this. A file with the name *filename* is made executable by the command *chmod x filename* (the command *man chmod* and reference 2 give more information about the command *chmod* used with different flags).

As an example the file `delete` in `/net/triton_disk2/user/bohms` contains the lines:

```
#!/bin/ksh  (which means that the file is executed in a program called
             kornshell)
stty intr ^Y (sets the terminalemulators interrupt sign to ctrl-Y)
stty erase ^? (sets the terminalemulators erase sign to ctrl-?)
```

Ctrl-Y means the sign created when both the ctrl key and the letter y key are pressed at the same time.

When `delete` is given as a command these lines are executed with the result that the delete key can be used in the rest of the terminalsession.

The variable `PATH` contains information where the computer should search when one gives a command (see ref. 2). More information about shell programmes is given in ref. 2.

A shell variable is set by the command `variablename=value` followed by the command `export variablename` where `variablename` is the name of the variable and `value` its new value.

The command `env` shows all the variables set in `PATH`

A useful command is `grep` which search for a string in a directory, e.g. `grep string filename` search for the string `string` in the file `filename`, `grep string *`, search for the string `string` in all the files in the directory.

## 2 The code XPDP1

### 2.1 Introduction

XPDP1 is a bounded electrostatic code for simulating 1 dimensional Plasma Devices. It is written in C for Unix workstations with X-Windows. Three versions of XPDP1 exist on the HP computer:

1. The original version with the sourcefiles saved in the directory `/net/triton_disk2/user/bohms/pdp1source` and the original executable file with the inputfiles saved in `/net/triton_disk2/user/bohms/xpdp1exe` (and in `/net/triton_disk2/user/torven/xpdp1`).
2. The latest version of XPDP1 saved in the directory `/net/triton_disk2/user/bohms/xpdp2/xpdp1`.
3. The modified version of the latest XPDP1 version saved in the directory `/net/triton_disk2/user/bohms/xpdp2/modxpdp1`.

A list of all the routines which should be compiled and linked together to make XPDP1 3.1 executable is given in appendix 7.

## 2.2 The creation of MODXPDP1 from XPDP1 3.1

The source code `pdp1.c` and some of the routines of XPDP1 version 3.1 has been modified to enable a nonuniform initial spatial particle distribution to be simulated. The modified version of XPDP1 3.1 is called MODXPDP1. In running MODXPDP1 the user has to decide if the initial particle density distribution at timestep = 0 should be uniform or nonuniform. The decision is made by setting the flag `densprofile` in the input file to 0 for a uniform or to 1 for a nonuniform initial density distribution. If one decide to have a nonuniform density distribution, a function describing the initial density distribution in space has to be given in a new subroutine. The routine is written only for symmetric maxwellian distribution but can easily be modified to include other distributions. The routine has to be added to the other routines in the compiling and linking process. Routines with different initial distribution has been given different names, but all have names beginning with `den` and are saved in the directory

`/net/triton_disk2/user/bohm/xpdp2/xpdp1/newfiles`. In the modified code it is also possible to simulate particles in the simulation region to be lost with a prescribed probability. The parameter `lossprob` (also read from the input file) is used in the subroutine `modpadjus.c`. This parameter determines the probability for a particle to disappear from the simulation region. If `lossprob` is set to 0 no ionlosses will be introduced.

A list of the files in this directory is given in appendix.

Some minor changes from the original code `pdp1.c` has also been made with the diagnostic tools. Three probes measuring the time variation of the electric field at given space coordinates and a probe measuring the timevariation of the electric potential has also been incorporated in the program as well as plots of the velocity distribution function at the space coordinate given in the inputfile.

## 2.3 How to handle `xpdp1.c`

The source code `pdp1.c` is most easily compiled and linked by the makefile which was distributed with the `pdp1` programpackage. Writing the command `makefile` (if you have the makefile in your directory) compiles and links `pdp1.c` and the accompanying routines (all saved in the directory `/net/triton_disk2/user/bohm/xpdp2/xpdp1`) The makefile does the linking process (or compiling process) only if any of the source files (or objectfiles) has been changed since the last use of the makefile.

Let us suppose that the name of the executable program of the source program `pdp1.c` is `xpdp1` (The name is given in the makefile). Observe that an executable program called `xpdp1` (version 3.0 of **XPDP1**), exists somewhere in the directory tree and the command `xpdp1` would hence run this version.

Once you are in the directory with the executable program `xpdp1`, you can run the program by writing its name followed by the input file: `xpdp1 inputfile` where *inputfile* is the file (usually with the extension `.inp`) from which the main program read all the inputparameters. This file doesn't have to be compiled but one can just change the parameters in the file using an editor and the

inputfile can immediately be used together with the executable program again. Appendix 6 shows an example of an input file. If the code has been runned earlier to a timestep with the results dumped into a file with the extension .dmp, it can be restarted from the timestep by the command: *xpdp1 finputfile.inp dumpfile.dmp* (where you replace dumpfile by the name of the dumpfile).

If the inputfile is not in the same directory as the executable file, the path to the inputfile must be specified.

As an example:

```
xpdp1 /net/triton_disk2/user/bohm/xpdp2/xpdp1/inp/vc.inp
```

runs xpdp1 with the file vc.inp situated in the directory  
/net/triton\_disk2/user/bohm/xpdp2/xpdp1/inp.

Several inputfiles, each containing parameters to simulate a specific physical model, are delivered with the original version of XPDP1 3.1. These files are saved in the directory

**/net/triton\_disk2/user/bohm/xpdp2/xpdp1/inp.**

The new inputfiles saved in the directory

**/net/triton\_disk2/user/bohm/xpdp2/xpdp1/newinp**

can be run with the modified version of XPDP1 3.1 which is called MODXPDP1. A list of these files are given in appendix. You have to specify the path to the inputfile if the executable file of MODXPDP1 is saved in another directory than the inputfile. Suppose you want to run MODXPDP1 (which of course have to be in the directory you are in for the moment, write *ls* and see) and the inputfile with the parameter you want to run are in another directory named

**/net/triton\_disk2/user/bohm/xpdp2/xpdp1/inp**

then write

```
modxpdp1 /net/triton_disk2/user/bohm/xpdp2/xpdp1/inp/inputfile
```

where instead of inputfile you give the name of the inputfile.

You can also either copy the executable program to the inp directory or you can copy the input file you want to the directory where the executable program is saved.

**Important!** Don't forget to make the necessary changes in the original inputfiles if you want to run **modpdp1.c**, which is the edited version of the original pdp1.c and which contains a lot of changes (specified below). You have to add eighth parameters in the original input files (which lie in the directory **/net/triton\_disk2/user/bohm/xpdp2/xpdp1/inp**), one flag for the initial particle density distribution, one flag for ionlosses, three for the coordinates (in units of gridcellnumber) of the three probes measuring the electric field, one for the coordinate of the probe measuring the voltage and two for the velocity boundaries Vmin and Vmax of the velocity distribution function. Vmin and Vmax should be given for all the species included. Compare with the inputfiles in the directory **/net/triton\_disk2/user/bohm/xpdp2/modxpdp1/newinp**.

A control panel will pop up on the upper left of the screen when the program is started (fig. 2). A menu of options is displayed in the upper left corner of the window. The program is started by clicking on the button Run. The value of a timestep is given as a parameter in the inputfile. The time is showed in the uppermost button. The simulation can be stopped at an arbitrary time by clicking on the button Stop. The simulation is restarted again by clicking on Start.



Clicking on the save button dump the current values on a file. The default file name will be the same as the inputfile but with the ending *dmp*. The program will then be restarted at the correct timestep by the command *xpdp1 inputfile.inp inputfile.dmp* where *inputfile* is the name of the inputfile.

Observe that, when a run is restarted from a dumpfile, the timehistory plots will start from the timestep the resultfile were dumped and not from zero timestep.

## 2.4 Problems in running XPDP1

Some problem can emerge in running XPDP1.

Problems will most certainly occur in big runs when the number of particles and gridcells are near the upper limit of the available memory. The program can then stop by several reasons:

1. The program can stop with the message: Bus error(coredump) if too much memory is used for the icons. This is overcome by changing appropriate window options in the file *initwin.c* at the commandline *XGSet2D* from open to iconic.
2. The program can stop after some timesteps with the message: **ADJUST: too many particles, species** followed by the number of the species. This number, usually 0 for electrons and 1 for ions, is given in the inputfile. This error can be overcome either by decreasing the influx of the particles given by **j0l** and **j0r** in the inputfile or by increasing the memory for the particlearrays. This is done by increasing the parameter **max np** in the inputfile. The maximum number of particles is about 2000000 if the run include two species and the number of gridcells is 2400. One timestep can then take nearly 3 minutes even if you are the only user on the HP computer.
3. The windows for the diagnostic tools are shown on the screen either as open windows or as small icons. The icon is marked as a small rectangle filled with four small squares. The icon is transformed to an open window if you click twice on it. If you click once on it a small window with the options Move, Size,.....Close will pop up. The same window pop up if you clicks once at the upper left linedenoted small square of a window.  
By clicking on the option Close, the program will be killed and the computer message will be:

XIO: fatal IO error 32 (Broken pipe) on X server "130.237.45.16:0.0"

after 4637 requests (4637 known processed) with 0 events remaining.

The connection was probably broken by a server shutdown or KillClient.

3. The program stop temporarily when you measure with the crosshair on a graph and you have to click on the crosshair once again to start the program again.
4. It has happened that a stopped run of *xpdp1* vanished when I clicked on the icon for the *Vx-X*

Phase space. After some time all the windows belonging to xdp1 disappeared and the message: Bus error (coredump) appeared on the screen. The run was very memory consuming with 2 species, each with nearly 2 million particles and the simulation region consisted of 1200 gridcells.

## 2.5 Diagnostics in XPDP1

A large set of diagnostic tools appear on the screen when XPDP1 is runned. The manual belonging to the X-Window version of PDP1 [4] doesn't describe the diagnostic tools but the user is referred to the manual for the PC version of XPDP1 [9].

iconname	physical quantity
f(E) Mid i	<p>The distribution function for species number i (the different species is given numbers, species 1 can for instance be electrons and species 2 ions).</p> <p>The distribution function is calculated over a region in space given by the parameters XStart and XFinish. These parameters and the parameters Emin, Emax and nbin, which define the boundaries for the energy distribution and the number of cells for the same, are read from the inputfile. The x value on the graph denote the number of particles in the surrounding cell and the y value denote the number of particles divided by the squareroot of the energy. The y value in the one dimensional case will then be the number of particles between <math>v_x</math> and <math>v_x + dv_x</math>.</p>
f(E) i	Is the energy distribution function at left wall.
Mid Potential(t)	Shows the timehistory of the potential at the midplane of the system.
LHS Potential(t)	Shows the timehistory of the potential at the left electrode.
Local Mid Potential(t)	Shows the timehistory of the potential at the midplane of the system. The time is set to zero every nfft timestep.
Local LHS Potential(t)	Shows the timehistory of the potential at the left electrode. The time is set to zero every nfft timestep.
Time Average Potential(x) Time Average E field(x) etc.	The number of timesteps used to calculate the average of a physical quantity is set by the parameter nfft in the inputfile.

### 3 The VXT 2000 Windowing Terminal and its environment

#### 3.1 Problems with the terminal

If the terminal seems to be hanging you can reboot by pressing the keys 'Ctrl', 'Alt Function' and '.' on the same time. The '.' button should be the button on the numeric keypad which is the keypad to the right of the main keypad.

If the computer respond to a command by the message **cannot fork: too many processes**, too many old processes still exist. The command: `ps -ef` will show all the processes started by the user (and by other users). Kill old processes by the command `kill -9 pid` where instead of pid you put the identification number of the process (which will be given by the command `ps -ef`).

The command `rcmd luna xterm` should open a window on the computer named luna. One answer from the computer could be:

**luna being added to access control list**

**Xlib: connection to "triton:0.0" refused by server**

**Xlib: Client is not authorized to connect to Server**

**Error: Can't open displ: triton:0.0**

With this answer you have to speak with the system manager.

#### 3.2 How to print your files and graphs

A file consisting of readable lines is called a textfile. A text file should be sent to a text queue when it is to be printed. A postscript file is a file consisting of commands which is meant for an interpreter and which describes how, e.g. a graph, should be printed. If you run XPDP1 and want a graph to be printed you will click on the text 'print' above the graph. The instructions for printing the graph will then be saved in a postscriptfile with the ending .ps for postscript.

To print a file use the command `lp` together with a filedependent flag:

command given by the user	what will happen
(replace filename with the name of the file)	

\$ <code>lp -dansi filename</code>	A textfile will be printed from plafys laserwriter
\$ <code>lp -dtext filename</code>	A textfile                   "
\$ <code>lp -dpostscript filename</code>	A postscriptfile           "

\$ *lp -dsysprint* filename    A textfile will be printed from plafys linewriter  
 \$ *lp -dhp7550a* filename    A textfile will be printed from the local HP plotter  
 \$ *lp -dlps17* filename       A textfile will be printed from the lpsd\_17 plotter  
 \$ *lp -dlps17\_d* filename    A textfile will be printed on both side of the sheet on the lpsd\_17 plotter.

Plafys laserwriter is situated on the third floor in the corridor just outside room nr 2520. The lps17\_d plotter, the plafys linewriter and the local HP plotter are all situated in the computer room on the third floor (room nr 2524).

It is very important to put a file in the correct queue e.g. a postscriptfile must be printed with the command *lp -dpostscript* and **not** with *lp -dtext*. A postscriptfile printed as a textfile gives a very long unreadable printing. If you by mistake put a file in wrong queue you must log in to the VAX computer and give the commands:

*show queue postscript*        which shows the jobs in the postscript queue  
*show queue text*                which shows the jobs in the text queue  
*show queue postscript/all*    gives all the jobs including other users.

the computer responds with the entrynumbers for the different printingjobs.

*delete/entry* = entrynumber

will then stop the printingjob.

If you want to set a job in another queue you should give the command

*set entry* entrynumber */requeue*=newqueueusername where instead of entrynumber and newqueueusername you put the number of the job and the name of the new queue e.g. ps.plot.

The plotter lps17\_d has the advantage that you don't have to specify the kind of file you plot.

A postscriptfile will be plotted as a postscriptfile and a textfile will be plotted as a textfile.

One problem with this plotter is that you can not send to big files or to many files to this plotter at the same time because it's limited memory.

Warning! If you try to print a file which is too big an error will appear and the queue to the printer will be stopped.

## References

- [1] Waite, M., Prata, S., Martin, D., Programming i C, PAGINA förlags AB.
- [2] Användarhandledning för UNIX av Berkeley-typ, Systemgruppen, Tekniska högskolan i Stockholm, Numerisk Analys och Datalogi.
- [3] Kernighan, B., Ritchie, D., The C programming language, Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632.
- [4] Vahedi, V., Verboncoeur, J., XPDP1 Plasma Device 1 Dimensional Bounded Electrostatic Code, Reference manual XPDP1 version 3.1, c/o Prof. C.K. Birdsall, Plasma Theory and Simulation Group, Electronics Research Laboratory, Cory Hall, University of California, Berkeley, CA 94720
- [5] Verboncoeur, J., Alves, V., Vahedi, V., Simultaneous Potential and Circuit Solution for Bounded Plasma Particle Simulation Codes. Memorandum No. UCB/ERL M90/67, 7 August 1990.
- [6] Lawson, W.S., Particle Simulation of Bounded 1D Plasma Systems, Electronics Research Lab, University of California, Berkeley, California, Journal of Computational Physics, Vol. 80, No.2, Feb. 1989.
- [7] Birdsall, C.K., Particle-in-Cell Charged-Particle Simulations, Plus Monte Carlo Collisions With Neutral Atoms, PIC-MCC, IEEE Transactions on Plasma Science, Vol. 19, No. 2, April 1991
- [8] Alves, M.V., Lieberman, M.A., Vahedi, V., Birdsall, C.K., Sheath Voltage Ratio for Asymmetric rf Discharges.
- [9] PDP1, PDC1, PDS1 Plasma Device 1 Dimensional Bounded Electrostatic Codes, Reference manual PC version 2.1, c/o Prof. C.K. Birdsall, Plasma Theory and Simulation Group, Electronics Research Laboratory, Cory Hall, University of California, Berkeley, CA 94720
- [10] Lawson, S.L., Gray, P.C., PDW1/PDW2 User's Manual, Aug 1989
- [11] Lawson, W.S., PDW1 User's Manual, Electronics Research Laboratory, Cory Hall, University of California, Berkeley, CA 94720



## Appendix 1

### An example of a terminalsession

A dialogue with the computer can be saved in the file *filename* with the command *script filename*. The recording is stopped with the command *exit*. Here follows a terminaldialogue.

The commands given by the user is given in italics. The response from the computer is written in bold.

*\$ delete*

Running the command *file delete* makes the delete key on the keyboard to work properly. A copy of the commando-file must however exist in the directory you are located in when you run the file.

*\$ pwd*

**/net/triton\_disk2/user/bohm**

Shows where you are located in the directory tree.

*\$ ls -l -a -t*

*ls* list the content of the directory. The flag *l* gives detailed information of the files and directories. The flag *a* includes all files in the list, also files preceded by a point.

```
total 62
-rw-r--r-- 1 bohm  ext   116 Apr  4 12:49 dialogue.tex
-rw----- 1 bohm  ext  3972 Apr  4 12:49 .sh_history
drwxr-xr-x 10 bohm  ext   3072 Apr  4 12:49 .
drwxr-xr-x  7 bohm  ext   1024 Apr  4 11:29 .vue
-rw----- 1 bohm  ext    49 Apr  4 11:28 .Xauthority
drwxr-xr-x  2 bohm  ext   1024 Mar 31 08:23 pdp1source
drwxr-xr-x  5 bohm  ext   1024 Mar 29 17:10 xdp2
dr-xr-xr-x  8 root  sys   1024 Mar 14 15:48 ..
-rwx----- 1 bohm  ext  2702 Nov 25 13:58 .hextris
drwx----- 2 bohm  mail  1024 Nov 24 15:52 .elm
drwxr-xr-x  2 bohm  ext   1024 Nov 15 14:25 xdp1exe
drwxr-xr-x  4 bohm  ext   1024 Nov  9 14:35 xdp1backup
drwx----- 2 bohm  mail    24 Nov  8 14:55 Mail
drwxrwx---  2 bohm  ext   1024 Aug 18 1994 .lrom
-rwxr-xr-x  1 bohm  ext    38 Aug 17 1994 delete
-rwxr-xr-x  1 bohm  ext  3969 Aug 17 1994 .vueprofile
-rwxr-xr-x  1 bohm  ext  4997 Aug 16 1994 .profile
```

*\$ cd xdp2*

Go to the subdirectory *xdp2* (which must be located in the same directory as you are in)

*\$ ls*

**modxpdp1 README xgrafx xdp1 xdp1.tar**

*\$ cd modxpdp1*

*\$ ls*

**makefile modinitwin.c modpdp1.h modxpdp1 newres  
mcc.c modpadjus.c modpflmv.c newfiles  
modfft.c modpdp1.c modprest.c newinp**

*\$ cd newfiles*

```
$ ls
den2test.c  den60test.c  den80test.c  densredcold.c  dentest.c
den3test.c  den6ctest.c  den90test.c  densredloss.c  exptest.c
den4test.c  den6test.c  densloss.c  densred.c  remember.tex
den5test.c  den7test.c  densnew.c  dens.c
```

\$ cp den7test.c .. Copy the file den7test.c to the directory one level up.

\$ cd .. Go one directory level up.

\$ edit makefile Edit the file makefile.

"makefile" 43 lines, 1345 characters

:/den Find the string den (the colon is the prompt given in the edit mode).

PDP1OBJ=modfft.o modpdp1.o modprest.o modpflmv.o modpadjus.c den7test.o\

:q Leave the edit mode.

\$ make Run the makefile.

cc -c -I/usr/include modfft.c

cpp: warning: Cannot read american.iso88591 message catalog; using default language

cc -c -I/usr/include modpdp1.c

cpp: warning: Cannot read american.iso88591 message catalog; using default language

cc -c -I/usr/include modprest.c

cpp: warning: Cannot read american.iso88591 message catalog; using default language

cc -c -I/usr/include modpflmv.c

cpp: warning: Cannot read american.iso88591 message catalog; using default language

cc -c -I/usr/include den7test.c

cpp: warning: Cannot read american.iso88591 message catalog; using default language

cc -c -I/usr/include modinitwin.c

cpp: warning: Cannot read american.iso88591 message catalog; using default language

cc -c -I/usr/include mcc.c

cpp: warning: Cannot read american.iso88591 message catalog; using default language

cc -I/usr/include -o modxpd1 modfft.o modpdp1.o modprest.o modpflmv.o m

odpadjus.c den7test.o modinitwin.o mcc.o ./xgrafx/xgrafx.o -lX11 -lm -L/usr/i

nclude  
cpp: warning: Cannot read american.iso88591 message catalog; using default language

```
$ ls
den7test.c  modfft.c  modpdp1.c  modprest.c  newres
den7test.o  modfft.o  modpdp1.h  modprest.o
makefile  modinitwin.c  modpdp1.o  modxpd1
mcc.c  modinitwin.o  modpflmv.c  newfiles
mcc.o  modpadjus.c  modpflmv.o  newinp
```

\$ modxpd1 newinp/test8a.inp Run modxpd1 with test6a.inp, situated in the directory newinp, as inputfile.

XPDP1 - Bounded Electrostatic 1 Dimensional Code

version 3.1

(c) Copyright 1988-93 Regents of the University of California

Plasma Theory and Simulation Group

University of California - Berkeley

See README file for information about Licensing

XGrafix version 1.0

(c) Copyright 1992 The Regents of the University of California  
Authors: Vahid Vahedi and John Verboncoeur

XGrafix is running on DECWINDOWS DigitalEquipmentCorp. / VXT 2000 version 2001 o  
f the X Window System, X11 R0  
Color plane depth ..... 8  
Display width ..... 1280  
Display height ..... 1024  
The display 130.237.45.16:0.0

\$ cd newinp

\$ ls

```
input.tex      provpotdrop.inp  test6cnograd.inp  testpotdrop.inp
ionlosses.inp  showpotdrop.inp  test6c.inp        test.inp
ionredlosses.inp test2.inp        test6.inp        vulcan.inp
multi1.inp     test3.inp        test7.inp
potdropred.inp test4.inp        test8a.inp
potdrop.inp    test5.inp        test8.inp
```

\$ edit test8a.inp

"test8a.inp" 69 lines, 2728 characters

:1

Go to line 1 in the file.

STUDIES OF ANOMALOUS POTENTIAL DROPS IN PLASMAS DUE TO ION DENSITY INHOMOGENEITY  
ES.

:

/\* Version 9500209 \*/

:

/\*\*\*\*\*\*nc2p=1.352e9 gives with fn1=9.6154e3 cm-1 wpe app. 2e9 \*\*\*\*\*/

:

-nsp---nc---nc2p---dt[s]---length[m]--area[m^2]--epsilon--B[Tesla]---PSI[D]--

:

2 800 8.675e8 0.25e-9 1e-1 1.0 1.0 0 0

:change

The command change means that the line should be changed.

2 x 8.675e8 0.25e-9 1e-1 1.0 1.0 0 0 Write the new line.

The '.' means that the change is completed.

:wq

Write the changed file over the old one and leave the edit mode.

"test8a.inp" 69 lines, 2728 characters

\$ cd ..

\$ modxpdpl newinp/test8a.inp

If you now run the program it will be locked and will not continue.  
Do Ctrl Y and you will leave the program.

XPDP1 - Bounded Electrostatic 1 Dimensional Code

version 3.1

(c) Copyright 1988-93 Regents of the University of California  
Plasma Theory and Simulation Group  
University of California - Berkeley

See README file for information about Licensing

\$

\$ exit

Close the window.

## Appendix 2

Advices (in swedish) from the system manager.

Tips för HP-UX datorerna

Datum 950227

----- Alfvénlaboratoriets HP-UX datorer -----

För att göra det lite enklare att använda datorerna finns numera en "tipsfil" med namnet,

/net/luna\_disk1/scripts/user\_info

Denna fil börjar med en innehållsförteckning, det är meningen att den ska innehålla tips om Mentor, SDRC och operativsystemet. Vill du ha den på papper så skriv,

```
$ lp -dtext /net/luna_disk1/scripts/user_info
```

Om du ska läsa den och använder HP-VUE klicka på anteckningsblocket (till höger om terminalen), under "file" väljer du sedan "open" och letar dig fram till filen.

Den kan också läsas ifrån en terminal med tex kommandot "more",

```
$ more /net/luna_disk1/scripts/user_info
```

I "more" stegar du fram till nästa sida med mellanslag och avslutar med q.

Om DU har synpunkter eller tips att dela med dig av, hör av dig till mig.

Med vänliga hälsningar,

Lasse Bylander

## Appendix 3

Instructions from the file README (in the directory `/net/triton_disk2/user/bohm/xpdp2`) how to make a tar file uncompressed to readable files. A tar file (usually ending with `.tar`) contains files which are compressed. This is a way to make files smaller in size and hence more transportable.

XPDP1 version 3.1

(c) Copyright 1991-1993 The Regents of the University of California

Steps to create the xpdp1 executable:

1) `uncompress xpdp1.tar.Z`

This will uncompress the distribution file. Then type

2) `tar xpvf xpdp1.tar`

to extract the files from the tarfile. This will create the `xgrafx` and `xpdp1` directories.

3) `cd xgrafx`

4) `make`

This will create the `xgrafx.o` file which is the GUI for `xpdp1`. You might have to change the makefile if your X11 libraries are in a non-standard place.

5) `cd ../xpdp1`

6) `make`

This will create the program called `xpdp1`.

--Edited 4.7.93 by Payam Mirrashidi



## Appendix 4

A version (den6test.c) of the routine specifying an initial nonuniform density distribution. The file has to be compiled and linked with modpdp1.c.

```

/* Version 950118 */
/* DENS so far only for thermal symmetric distribution */
#include "modpdp1.h"
dens(isp,k2,vel)
/* The variables used as arguments are declared before the subroutine start */
int k2;
int isp;
float vel[];
/* The start of the subroutine is marked with { and the end of the subroutine with } */
{
char *malloc();
float fn1,Dn;
float Y0,rid,X0,ris,a,az,z1;
float sq;
/* A static variable keeps its value even when the program leaves the subroutine
*denspek is declared as a pointer. */
static int *denspek;
float theta,vmag;
int kkr1;
float pix, del, zL, z, ri;
int kkom;
int kkr, iki, ik, iaa, ix, i;
/* Notice that kom is declared in modpdp1.h */
/* del is the number of centimeter/gridcell */
del=length*100./nc;
zL=6.590574843;
kom=0;
/* The total number to be loaded is calculated just once: for k2=0 and for
species=0. */
if(!k2)
{
if(!isp)
{
/* Allocate space for pointer denspek */

```

```

denspek=(int *)malloc(4*ng);
kkr1=0;
fn1=9615./2.;
Dn=39903;
z1=1.48560366;
X0=10.;
ris=10.*fn1-9.5*fn1*sin(PI*z1/zL+PI/180.*76.73732399)*sin(PI*z1/zL+PI/180.*76.73
732399);
rid=-9.5*fn1*PI/zL*sin(2*(PI*z1/zL+PI/180.*76.73732399));
Y0=(3.*ris*ris-10.*rid*ris)/(2.*ris-10.*rid);
b=fabs(2.*ris-Y0);
a=fabs(sqrt(100.*(2.*ris-Y0)*(2.*ris-Y0)/((2.*ris-Y0)*(2.*ris-Y0)-(ris-Y0)*(ris-
Y0))));
/* iki is the number of gridcell and iki=0 the cathode */
for(iki=0; iki<ng; iki++)
{
/* z is the distance in centimeter for the gridcells */
z=iki*del;
/* ri is the number of superparticles/centimeter at the gridcells */
/* different functions for the density in different regions */
if(iki<=119)
ri=10.*fn1-9.5*fn1*sin(PI*z/zL+PI/180.*76.73732399)*sin(PI*z/zL+PI/180.*76.73732
399);
else
if(iki>119 && iki<=919)
{
az=(z-10.)*(z-10.)/(a*a);
ri=b*sqrt(1.-(z-11.4375)*(z-11.4375)/(a*a))+Y0;
}
else
/*ri=10*(5.*fn1-4.5*fn1*sin(PI*0.4375/zL+PI/180.*70.5288)*sin(PI*0.4375/zL+PI/18
0.*70.5288));*/
ri=2.*ris;
/* printf("\n%10f",del); */
/* kkr is the number of superparticles/gridcell */
kkr=ri*del+0.001;
/* Because the DENS routine is called twice for each species from the routine
LOAD in the main programme, kkr, the total number to be loaded for each species
is divided by two. */
/* At the boundaries the number of particles is reduced by two because the
length of each cell surrounding each gridpoint at the boundary will be half the
gridcell length. */
/* The variable pix will shift the coordinates of the particles at the boundari-
```

```

ries a little bit toward the centre of the simulation region so they aren't
contributing to the charges on the endplates. */
/*  printf("\n%10d",iki); */
/*  printf("\n%10d",kkr); */
    if(iki==0||iki==nc)
kkr=kkr/4+0.1;
    else
kkr=kkr/2+0.1;
    *(denspek+iki)=kkr;
    kkr1=kkr1+kkr;
}
}
}
/*  printf("\n%10d",kkr1); */
    if (2*kkr1 > maxnp[isp])
    {
        puts("DENS : too many particles, species ");
        putchar(isp+49);
        exit(1);
    }
    for(ik=0; ik<ng; ik++)
    {
        if(ik==0||ik==nc)
        pix=0.0002*(1.-2*ik/nc);
        else
        pix=0;
    for(iaa=0; iaa< *(denspek+ik); iaa++)
    {
        ix=kom+np[isp];
        x[isp][ix]=ik+pix;
        vx[isp][ix] = *(vel+Index[(kom+k2)%1024]);
        vmag= vperp0[isp]+vperpt[isp]*sqrt(2*fabs(log(frand()+DBL_MIN)));
        theta= TWOPI*frand();
        vy[isp][ix] = vmag*sin(theta);
        vz[isp][ix] = vmag*cos(theta);
        kom=kom+1;
    }
    }
    if(k2)
    {
        if(dde)
        {
            kkom=np[isp]+kom;

```

```

for(i=0; i<=kkom; i++)
{
  ix = np[isp] +i;
  x[isp][ix] += dde*sin(TWOPI*x[isp][ix]/xnc);
}
}
return;
} /* end of density */

```

## Appendix 5

The new inputfiles which should be run with modpdp1.c which is the new version of pdp1.c. The files are saved in the directory /net/triton\_disk2/user/bohm/xpdp2/xpdp1/newinp.

<u>inputfile</u>	<u>included parameters</u>	<u>remark</u>
test8a.inp	all	Ready to use with modxpdp1.
multi1.inp	all except $V_{\min}$ and $V_{\max}$	
test6cnograd.inp	”	
test8.inp	”	
test6c.inp	”	
test7.inp	”	
test6.inp	”	
test5.inp	all except xefieldx3, $V_{\min}$ and $V_{\max}$	
test4.inp	”	
test3.inp	”	
potdrop.inp	all except xefieldx3, xefieldx2, xpotfield, $V_{\min}$ and $V_{\max}$	Runned with dens.c gives a potential- drop and a region with beamplasma interaction.
showpotdrop.inp	”	Gives a show of beamplasma interaction. Use with densred.c.
provpotdrop.inp		
test2.inp	all except xefieldx3, xefieldx2, xpotfield, $V_{\min}$ and $V_{\max}$ but have the	

	extraparameter phasenum	
test.inp	"	Use with dentest.c
testpdrop.inp	all except xefieldx3, xefieldx2, xefieldx1, xpotfield, $V_{\min}$ and $V_{\max}$	
vulcan.inp	"	Cold particles. Use with densredcold.c.
potdropred.inp	"	The same as pdrop.inp but reduced the particledensity by 10. Use with densred.c.
ionlosses.inp	"	Use with densloss.c
ionredlosses.inp	"	The same as ionlosses.inp but reduced the particledensity by 10. Use with densredloss.c.

## Appendix 6

The inputfile test8a.inp which include all the parameters to be runned with modpdp1.c.

STUDIES OF ANOMALOUS POTENTIAL DROPS IN PLASMAS DUE TO ION DENSITY INHOMOGENEITIES.

/\* Version 9500209 \*/

/\*\*\*\*\*\*nc2p=1.352e9 gives with fn1=9.6154e3 cm-1 wpe app. 2e9 \*\*\*\*\*/

-nsp---nc---nc2p---dt[s]---length[m]---area[m^2]---epsilon---B[Tesla]---PSI[D]--  
2 800 8.675e8 0.25e-9 1e-1 1.0 1.0 0 0

-rhoback[C/m^3]---backj[Amp/m^2]---dde---extR[Ohm]---extL[H]---extC[F]---q0[C]--  
0.0 0.0 0.0 0.0 0.0 1e18 0.0

-dcramped---source---dc[V/Amp]---ramp[(V/Amp)/s]---ac[V/Amp]---f0[Hz]---theta0[D]-  
1 v -56.85 -1e18 0.0 0.0 0.0

-secondary-e\_collisional-i\_collisional-reflux-nfft-nsmoothing-ntimestep-denspro  
file-lossprob-xefieldx1[gridcellnumber]-xefieldx2-xefieldx3-xpotfield  
0 0 0 0 256 2 0 1 0 431 468 493 93

--seec(electrons)---seec(ions)---ion species---Gpressure[Torr]---GTemp[eV]---  
0.0 0.0 0 0 0

## ELECTRON-NEUTRAL COLLISIONAL PARAMETERS-----

```
--selsmax[m^2]---elsengy0[eV]---elsengy1[eV]---elsengy2[eV]-
1.0e-19    0.0      0.0      10.0
```

```
--sextmax[m^2]---extengy0[eV]---extengy1[eV]---extengy2[eV]---
1.0e-20    12.0     50.0     100.0
```

```
--sionmax[m^2]---ionengy0[eV]---ionengy1[eV]---ionengy2[eV]---
1.0e-20    13.6     60.0     110.0
```

## ION-NEUTRAL COLLISIONAL PARAMETERS-----

```
--achrgx[m^2]--bchrgx[m^2/V^1/2]-----ascat[m^2]--bscat[m^2/V^1/2]---
3.0e-19    0.0      2.0e-19    0.0
```

## SPECIES 1

```
----q[C]-----m[Kg]---j0L[Amp/m^2]---j0R[Amp/m^2]----initn[m^-3]--
-1.602e-19  9.11e-31  166.167    166.167    1
```

```
--v0L[m/s]---v0R[m/s]---vtL[m/s]---vtR[m/s]----vCL[m/s]---vCR[m/s]-
0.0    0.    1.0e6    1.0e6    0.    0.
```

```
---vperpt[m/s]---vperp0[m/s]---nbin----Emin[eV]----Emax[ev]---max-np--
0.0    0.0    100    0.    100. 300000
```

```
-For-Mid-Diagnostic-nbin-Emin[eV]-Emax[eV]-Vmin[m/s]-Vmax[m/s]-XStart-XFinish-
100 0.0    100.0 -1000000 1501000 .0435 .0445
```

## SPECIES 2

```
----q[C] -----m[Kg]---j0L[Amp/m^2]---j0R[Amp/m^2]----initn[m^-3]--
1.602e-19  6.68139e-26  0.33607    0.33607    1
```

```
--v0L[m/s]---v0R[m/s]---vtL[m/s]---vtR[m/s]----vCL[m/s]---vCR[m/s]-
0.    0.    2.0225e3  2.0225e3    0.    0.
```

```
---vperpt[m/s]---vperp0[m/s]---nbin----Emin[eV]----Emax[ev]---max-np--
0.0    0.0    100    0.    100. 300000
```

```
-For-Mid-Diagnostic-nbin-Emin[eV]-Emax[eV]-Vmin[m/s]-Vmax[m/s]-XStart-XFinish-
100 0.0    100.0 -1000000 1501000 .0435 .0445
```

## Appendix 7

List of the routines which should be compiled and linked together to make XPDP1 executable:

argonmcc.c

fft.c

initwin.c

They differ in the initial density distribution.

size (in bytes)	last modification	filename	remarks
3318	Feb 28 17:32	den80test.c	
3134	Feb 23 16:48	den60test.c	
3689	Feb 21 17:59	den6ctest.c	
3605	Jan 31 19:33	den7test.c	
3670	Jan 20 12:03	den6test.c	
3816	Jan 16 16:25	den5test.c	
3403	Dec 23 10:06	den4test.c	
3097	Dec 22 13:04	den3test.c	
3054	Dec 21 16:20	den2test.c	
3221	Dec 13 08:39	dentest.c	
3132	Dec 8 16:39	densnew.c	
3024	Dec 8 13:19	dens.c	potentialdrop with beamplasmaregion
3024	Nov 28 09:50	densred.c	" reduced the particle density by 10
3374	Nov 28 09:50	densredcold.c	cold particles
2991	Nov 28 09:50	densloss.c	potentialdrop with or without ionlosses
2991	Nov 28 09:50	densredloss.c	" reduced the particle density by 10



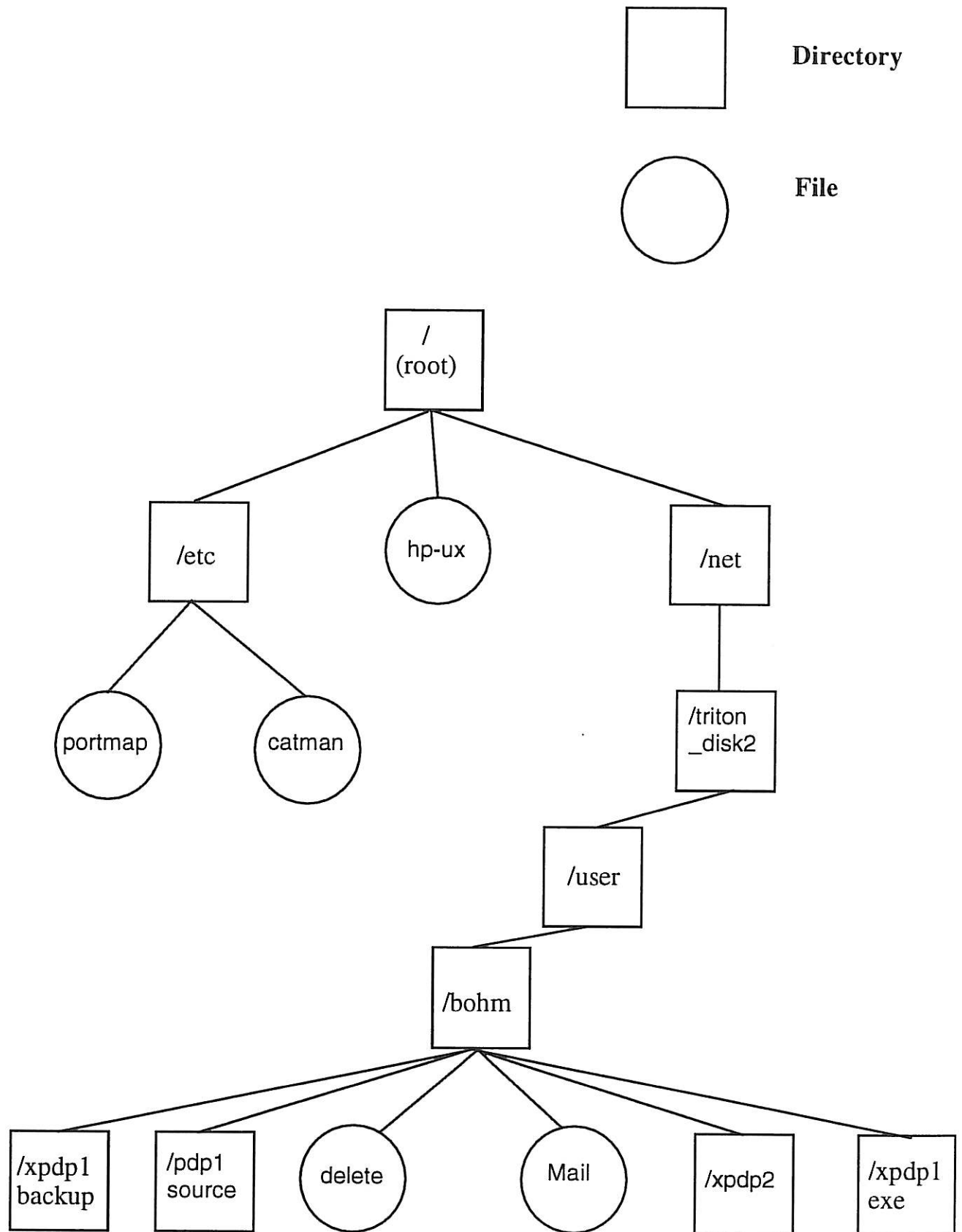


FIG. 1. An example of a directory tree

- xdp3 Control Panel • □
Time = 0
Run
Stop
Step
Save
Quit
Special

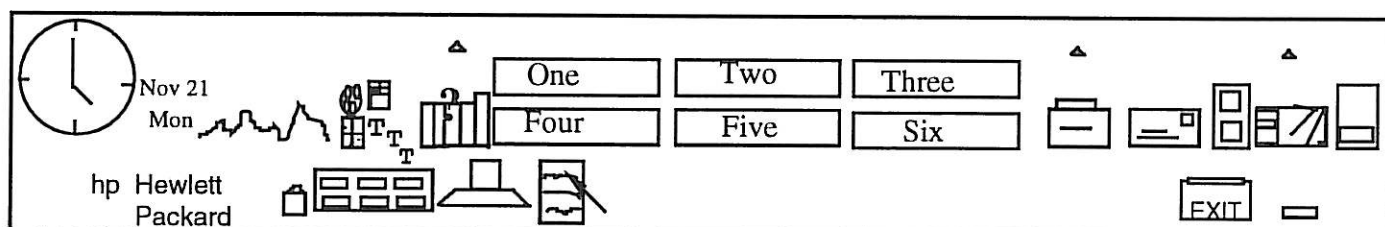
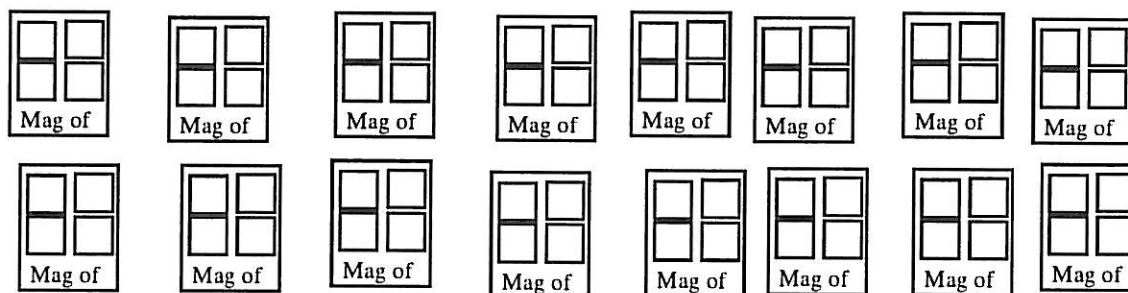
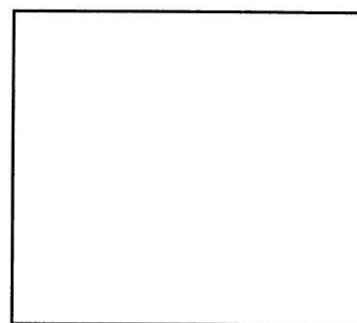
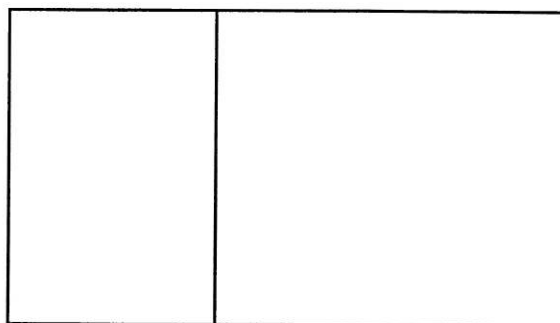


FIG. 2. The appearance of the screen in running XPDP1 and MODXPDP1.

